Journal of Symbolic Computation 47 (2012) 589–609

Contents lists available at SciVerse ScienceDirect

Journal of Symbolic Computation

journal homepage: www.elsevier.com/locate/jsc

Transforming problems from analysis to algebra: A case study in linear boundary problems

Bruno Buchberger [a], Markus Rosenkranz [b]

[a] Research Institute for Symbolic Computation (RISC), Johannes Kepler Universität, Altenberger Str. 69, 4040 Linz, Austria

[b] School of Mathematics, Statistics and Actuarial Science (SMSAS), University of Kent, Canterbury CT2 7NF, United Kingdom

ARTICLE INFO

Article history:
Received 10 September 2011
Accepted 30 October 2011
Available online 24 December 2011

Keywords:
Symbolic analysis
Linear boundary problems
Wen-Tsun Wu
Wolfgang Groebner
Theorema
Algorithmic functors
Groebner bases
Integro-differential algebras
Baxter algebras

ABSTRACT

In this paper, we summarize our recent work on establishing, for the first time, an algorithm for the symbolic solution of linear boundary problems. We put our work in the frame of Wen-Tsun Wu's approach to algorithmic problem solving in analysis, geometry, and logic by mapping the significant aspects of the underlying domains into algebra. We briefly compare this with the lines of thought of Wolfgang Groebner. For building up the necessary tower of domains in a generic and flexible way, we use the machinery of algorithmic functors introduced in our Theorema project. The essence of this concept is explained in the first section of the paper.

The main part of the paper then describes our symbolic analysis approach to linear boundary problems, which hinges on three basic principles: (1) Differentiation as well as integration is treated axiomatically, setting up an algebraic data structure that can encode the problem statement (differential equation and boundary conditions) and suitable symbolic expressions for their solution (Green's operators qua integral operators). (2) Abstract boundary problems are introduced as pairs consisting of an epimorphism on a vector space (abstract differential operator) and a subspace of its dual (abstract boundary conditions). (3) Operator algebras are treated by noncommutative polynomials, modulo Groebner bases for certain relation ideals.

E-mail addresses: bruno.buchberger@risc.jku.at, Bruno.Buchberger@risc.uni-linz.ac.at (B. Buchberger), M.Rosenkranz@kent.ac.uk (M. Rosenkranz).

URLs: http://www.risc.uni-linz.ac.at/people/buchberger (B. Buchberger), http://www.kent.ac.uk/ims/personal/mgr/index.html (M. Rosenkranz).

0747-7171
doi:10.1016/j.jsc.2011.12.022

1. Introduction

Professor Wen-Tsun Wu, by his oeuvre, marked a significant turning point in 20th century mathematics: The introduction of *algorithms in abstract algebra, analysis, and logic*.

Mathematics in the 19th century excelled in creating a wealth of new notions, theorems, and methods for difficult problems in all areas of mathematics (mainly motivated by problems in the natural sciences). However, 19th century mathematics did not yet work out the abstract, structural, axiomatic aspects of mathematical theories, it did not yet clearly understand the logical foundation of mathematics, and it had no clear notion of algorithm. Rather, the mathematics of the 19th century was all about *concrete objects*, reasoning was intuitive, and methods were formulated by just giving typical examples.

In the 20th century, the *abstract, axiomatic approach* was introduced in mathematics, culminating in the Bourbaki effort for which Wen-Tsun Wu also made important contributions. At the same time – and in close interaction with the abstract, axiomatic approach – fundamental insight into the foundation of mathematics was gained in the emerging field of mathematical logic. In this context it appeared to be necessary to come up with a clear notion of algorithm, in other words, a notion of universal computing mechanisms (universal programmable computers). Interestingly, this happened quite some years before the advent of the first physical devices that made universal programmable computing possible. And also interestingly, while the fundamentally mathematical invention of computer (in the sense of universal programmable computing device) revolutionized science, technology, and practical life in the 20th century, abstract (“pure”) mathematics more and more forgot computing, up to the point that words like “algorithms”, “computers”, “efficiency”, *etc.* do not even occur in the Bourbaki collection. This is unfortunate: it had and has quite negative consequences for the role mathematics plays today in society, education, and politics.

One may of course reply that numerical mathematics was and is considered to be the computational outlet of mathematics and the bridge to engineering and real life. However, for many mathematicians, numerical mathematics is a compromise leading away from true mathematics by replacing the actual mathematical objects and domains by finitary approximations. In contrast, in our view, the *algorithmic treatment of mathematical problems* in the original, non-approximated, domains is the core of mathematical aspiration, which strives toward understanding a difficult problem so deeply that the infinitely many instances of the problem can be handled by a uniform “rule” (a theorem that has to be proved). However, how can problems in abstract mathematical structures, notably structures in analysis (in which we deal with uncountable sets of non-finitary objects like the field of real numbers or various function algebras) be turned into problems in algorithmic domains: domains consisting of countably many finitary (computer representable) objects with decidable membership and algorithmic functions and predicates on them?

The clue is that, instead of solving problems in the actual mathematical domains (which are essentially non-algorithmic), one considers *finitary representations* of these domains – meaning finitary object representations for countable subsets of the domain carriers – and one develops a mathematical theory that maps the operations in the original domains to algorithmic ones on the finitary representations. Typically, the theory necessary for that is quite demanding. It might seem that the inevitable limitation imposed by countable subsets of the abstract domains must lead to insignificant results. However, gathering more and more non-trivial theory will gradually expand the algorithmically subsets to cover more and more of the practically interesting cases.

For example, the problem of *indefinite integration* was first solved completely by Risch’ theory (Risch, 1969) for the countable domain of elementary transcendental functions (real functions representable by elementary transcendental expressions). The basic domain was later expanded by more and more additional functions, with a suitable theory built up along the way. The same approach can be repeated in the whole hierarchy of domains starting from basic algebraic ones like the rationals or algebraic numbers, continuing through function domains on such domains, then operator domains on the function domains, and so on.

In a sense, all of mathematics is made part of algebra by this technique of finitary representations of domains via symbolic expressions. This is also true about the algorithmization of the meta-level of mathematics, *i.e. logic*: For example, the most popular algorithm for automated theorem proving in

first-order predicate logic, Robinson's resolution method, basically turns proving first order formulas with quantifiers into solving systems of algebraic equations of a very general nature (clauses).

Main-stream mathematics in the 20th century, has turned away from algorithms. Main-stream computer science, to a great extent, has turned away from mathematics. And mathematical logic – despite its exciting results about the possibilities and limitations of automation in mathematical thinking – has had very little influence on the daily practice of the working mathematician. In these surroundings, *Wen-Tsun Wu*, set a brilliant example of bringing together structural mathematics (Bourbakism, based on set theory), mathematical logic and algorithmics, exploiting their dynamic interactions (Wu, 2008). He showed how structural mathematics (notably parts of the theory of differential equations) can be cast in algebraic terms accessible to algorithms, and he demonstrated how algebraization can make geometrical theorem proving algorithmic.

We like to compare this with the attitude and contribution of *Wolfgang Groebner* (1899–1980): His attitude and endeavor basically pointed into the same direction as Wu's with a noticeable difference in style that can be explained by his being twenty years older than Wu. He was heavily involved in the structural/set theoretic reformulation of algebraic geometry (polynomial ideal theory), and he had always viewed analysis (differential equations), algebra, and geometry (differential geometry) as closely related areas, in all of which he was a true master. His work has always been directed toward establishing methods (Lie series for the numerical treatment of differential equations; algebraic methods for what at his time was called main problem of polynomial ideal theory, which in retrospect can be seen as the problem of constructing what are now called Groebner bases). His approach to algorithms was still example-based but he became very excited, toward the end of his professional life, by the advent of computers that would make both numerics and algebraic algorithmics truly mechanized.

We are now at the verge of a new century. Of course, nobody knows where mathematics will be at the end of the 21st century. In our personal view, we believe in and work for a mathematics that combines the abstract/axiomatic/structural "pure" aspect with the algorithmic world (including computer science) by using formal logic as a working language. In this way the activity of *building up mathematical theories* (invention of notions, invention and proof of theorems, invention of problems, invention and verification of algorithms) is turned into a systematic, computer-supported process. This is the motivation for our TH∃OREM∀ system (Buchberger et al., 2006), which we think stands in the tradition of people like Wu, Groebner and many others but tries to go a significant step further in the algorithmization of the mathematical invention and verification process itself.

In this paper, we survey recent work within the TH∃OREM∀ group that has resulted in symbolic algorithms for linear boundary problems, which can be seen as a further step in the algorithmization of analysis by algebraization and as a contribution to the emerging field of what could be called *symbolic analysis*—that part of symbolic computation that deals with problems from analysis, notably differential equations (Seiler, 1997), (Grabmeier et al., 2003, Section 2.11); cf. also the eponymous workshops of the FoCM conference series (Cucker and Shub, 1997). Beyond differential equations, there are undoubtedly numerous other areas in (functional) analysis that would benefit from a new treatment and evolution in symbolic analysis.

One such area is *boundary problems* (Stakgold, 2000; Duffy, 2001), the next obvious candidate after differential equations. The symbolic analysis of boundary problems has just started: there is a large, virtually unexplored territory ahead of us. In particular, virtually all results so far are for linear problems (and also the present survey of our work is restricted to this case). Even for such problems, however, mathematical software systems like Mathematica and Maple deliver solutions only in special cases, without a clear specification of the solvable cases (see Example 14).

The symbolic treatment of (linear) boundary problems has a somewhat *functional analytic flavor*: The boundary conditions of a well-posed boundary problem serve to define the Green's operator, an integral operator that maps an arbitrary forcing function to the unique solution of the problem. Since this is a linear operator, just as the differential operator (the left-hand side of the differential equation) and the accompanying boundary operators (the left-hand sides of the boundary conditions), one is naturally led to consider operator algebras—but without a topology.

The research summarized here originated from a joint seminar between the TH∃OREM∀ group and the group of Professor Heinz W. Engl of the RICAM institute. In this seminar, we came across the

paper Helton and Wavrik (1994) in which it was shown how noncommutative Groebner bases can be used for simplifying formulas in operator model theory. However, in Helton and Wavrik (1994) it was not noticed that Groebner bases have in fact another – maybe more important – potential: solving by elimination. It turned out that the *solving aspect of Groebner bases* can be brought to bear on linear boundary problems if all the relevant basic operators (differential, integral, boundary) are characterized by their fundamental interactions (Rosenkranz et al., 2003). Not only some contrived solutions are found in this way but exactly the well-known Green's functions of the classical Sturm–Liouville theory (which is restricted to self-adjoint differential operators of second order). This came to us as a surprise and illustrates another aspect of the intrinsic power of Groebner bases.

In its current form, the whole method hinges on *one* particular Groebner basis, which can be considered as a kind of universal Groebner bases for the set of linear boundary problems. As a consequence, the solving aspect of Groebner bases was pushed back to the simplifying aspect (but we expect that the former will have a comeback for certain future applications of symbolic boundary problems). The *Groebner basis for linear boundary problems*, in essence, is an encoding of the axioms of integro-differential algebras (Definition 1) on the operator level, cast in a suitable (noncommutative) polynomial domain.

It is a non-trivial task to find a suitable tower of algebraic domains in which this Groebner basis lives naturally. This is the more technical aspect of our work. For this, a very general – and algorithmic – notion of *functor* is crucial (Buchberger, 2001, 2008), which we introduced in the TH∃OREM∀ system and which allow the construction of domains like the ones we need in the context of our approach to linear boundary problems. For the details of the hierarchy of domains and functors that make it possible to obtain the solutions of linear boundary problems as Green's functions by simplification modulo the Groebner basis, we refer to the recent thesis (Tec, 2011) by L. Tec of the TH∃OREM∀ group. In this paper, we explain the mathematical design of this approach (Sections 3–5) as well as its realization by the notion and implementation of domains and functors in the TH∃OREM∀ system (Section 2).

The work surveyed in this paper is based on the following *contributions* (of course we must restrict ourselves to the most important ones):

- Buchberger initiated and leads the TH∃OREM∀ project and, in particular, introduced TH∃OREM∀ *domains and functors*, notably the Groebner ring functor based on the notion of a reduction ring (Buchberger, 2001).
- Rosenkranz created the symbolic approach to boundary problems (Rosenkranz, 2003, 2005), introducing the notion of *integro-differential algebras* and their *operator rings* (see the beginning of Section 3) as well as a *solution algorithm*.
- Engl and Buchberger provided *valuable stimuli* at this early stage (Rosenkranz et al., 2003), especially in the context of the Helton/Wavrik paper (see above).
- Regensburger and Rosenkranz jointly developed and elaborated the symbolic approach to boundary problems from 2004 to 2010. The key results of this period are the *multiplication/factorization* of boundary problems (Rosenkranz and Regensburger, 2008b), *abstract* boundary problems (Regensburger and Rosenkranz, 2009a), and *integro-differential polynomials* (Rosenkranz and Regensburger, 2008a).
- Tec contributed the *detailed implementation* of the necessary domains and functors in her doctoral thesis (Tec, 2011), under the guidance of Buchberger, Rosenkranz, and Regensburger.
- Rosenkranz introduced the ring of *partial integro-differential operators* (Rosenkranz et al., 2009); work on this topic is ongoing (Section 5).

As indicated above, we see the symbolic analysis of boundary problems as a vast, as yet (almost) unexplored territory; we hope for contributions from a variety of different people and different approaches.

2. Domains, functors, categories

Functors are an elegant tool for making algorithms domain-dependent. In our view, a *functor* is any function that takes domains as arguments and produces other domains. A *domain* is a carrier together with a couple of operations on the carrier. In other words, a functor takes the carriers and

the operations of the input domains and defines the carrier and the operations of the resulting domain in terms of the carriers and the operations of the input domains. In the case where the definitions of the operations in the functor are algorithmic, the operations in the resulting domain are algorithmic provided the operations in the input domains are.

A functor can also be understood as a mechanism that transforms properties of the input domains into properties of the output domain. We call a collection of formulas that describe properties of domains a "category description". Correspondingly, a *category* is the collection of all domains (taken from some universe of domains) that satisfy a given category description. The notions of category and functor are of course closely related to the corresponding notions in category theory. However, here we do not impose any algebraic properties on these notions. Any properties of operations on carriers constitute a category and any function (in particular any algorithmic function) mapping domains (carriers and operations) to new domains is a functor.

For a given functor F, we can formulate *conservation theorems*. These are theorems of the following structure (brackets are used in THEOREM for denoting function application): If domains $U, V, \ldots$, are in categories $\mathcal{C}, \mathcal{D}, \ldots$ then $F[U, V, \ldots]$ is in category $\mathcal{H}$. (Example of a typical conservation theorem: If C is a ring then the polynomial domain over C is also a ring.)

Our THEOREM setting of the functor view and methodology is very similar to the view and methodology of functors in *Standard ML* (Milner et al., 1997) but differs from it in two important respects:

- It is more general because it allows much more general constructions of carrier sets and operations, including nonalgorithmic constructions using all definition mechanisms of mathematics.
- It allows the formulation of categories and conservation theorems within the same language, namely the THEOREM version of predicate logic, and also aims at providing automated reasoning facilities for conservation theorems (as part of a general methodology for automated proofs).

Technically, there are many different ways of defining domains. In most algebra books, a domain is a tuple consisting of a carrier set and a couple of operations (functions and predicates) on the carrier. However, for THEOREM, we have adopted a crucially different representation, which we found advantageous for algorithmic mathematics where we do not want to have infinite sets as basic objects. This representation, in a certain respect, follows the concept of *interpretation* in model theory: An interpretation assigns operations (*i.e.* functions and predicates) to operations symbols (*i.e.* function symbols and predicate symbols).

In more detail, we say that a domain U can be applied to an operation f yielding an operation $U[f]$ that can now be applied to a couple of arguments, say x and y, to obtain $U[f][x, y]$. It is clear that, for this formulation of domains, we need the possibility of *Currying* in our language since $U[f]$ is a function that can then be applied to arguments x, y. Furthermore, we allow the abbreviating two-dimensional notation

$$\underset{U}{f}\,[x, y]$$

for $U[f][x, y]$. The *carrier* of a domain is not described by a set but by a unary decision predicate that yields true or false depending on whether or not the input belongs to the carrier of D. Typically, we will denote this decision predicate by $\in$, so that

$$\underset{U}{\in}\,[x]$$

yields true or false depending on whether or not the input x belongs to U.

Note that functors – in addition to arguments that are domains – may also have a couple of *other input arguments* as, for example the number n of indeterminates for the functor that constructs the domain of polynomials from a given coefficient domain or the number n for the functor that constructs the n-ary Cartesian product of a domain, etc.

As an *example of a functor* let us formulate, in THEOREM notation, the functor DirectProduct that takes a domain U with operation + and forms the direct product of U with itself:

$$\begin{aligned}&\text{DirectProduct}[U] = \text{Functor}[N, \text{any}[X, x1, x2, y1, y2],\\ &\quad \underset{N}{\in}[X] \Leftrightarrow \Big(\text{IsTuple}[X] \ \wedge\ |X| = 2 \wedge\ \underset{U}{\in}[X_1] \wedge\ \underset{U}{\in}[X_2]\Big)\\ &\quad \langle x1, x2\rangle \underset{N}{+} \langle y1, y2\rangle = \langle x1 \underset{U}{+} y1, x2 \underset{U}{+} y2\rangle\\ &]\end{aligned}$$

This should be understood as follows: Assume that we have a domain U with the unary decision predicate $U[\in]$ and a binary function $U[+]$. Then DirectProduct[U], the application of the functor DirectProduct to the domain U, gives us a new domain, let us call it N for the moment, such that

$$\underset{X}{\forall}\Big(\underset{N}{\in}[X] \Leftrightarrow \Big(\text{IsTuple}[X] \wedge |X| = 2 \wedge \underset{U}{\in}[X_1] \wedge \underset{U}{\in}[X_2]\Big)\Big),$$

where the built-in predicate IsTuple decides whether a given object is a tuple, *i.e.* an object of the form $\langle a1, \ldots, a_n\rangle$ and $|X|$ denotes the length of tuple X. In words: We have $N[\in][X]$, meaning X belongs to N, iff X is a tuple of length 2 and $U[\in][X_1]$ as well as $U[\in][X_2]$, meaning the first and second component of X belong to U. The definition

$$\langle x1, x2\rangle \underset{N}{+} \langle y1, y2\rangle = \langle x1 \underset{U}{+} y1, x2 \underset{U}{+} y2\rangle$$

describes the fact that the operation $N[+]$ in the new domain N is defined componentwise on tuples, using the operation $U[+]$ in U on the components.

Note that the THEOREM∀ functor construct can be conceived as an abbreviation of a predicate logic formula with a *such a quantifier* so that no new inference rules for functors are necessary. For example, the above functor definition can be read as an abbreviation of the following formula:

$$\begin{aligned}&\text{DirectProduct}[U] = \text{such an } N \text{ that}\\ &\bigwedge \begin{cases} \underset{X}{\forall}\Big(\underset{N}{\in}[X] \Leftrightarrow \Big(\text{IsTuple}[X] \wedge |X| = 2 \wedge \underset{U}{\in}[X_1] \wedge \underset{U}{\in}[X_2]\Big)\Big)\\ \underset{x1,x2,y1,y2}{\forall} \langle x1, x2\rangle \underset{N}{+} \langle y1, y2\rangle = \langle x1 \underset{U}{+} y1, x2 \underset{U}{+} y2\rangle \end{cases}\end{aligned}$$

Here the big wedge is the THEOREM∀ notation for a multi-line conjunction.

We illustrate now for the above example functor how one can call the functor in a typical computation in the frame of a THEOREM∀ standard session. For this we introduce the *integers* as basic domain by an introduction functor (a functor with no arguments):

$$\begin{aligned}&\text{IntegerNumbers}[] = \text{Functor}[N, \text{any}[x, y],\\ &\quad \underset{N}{\in}[x] \Leftrightarrow \text{is-integer}[x]\\ &\quad x \underset{N}{+} y = x + y\\ &]\end{aligned}$$

Here is-integer and $+$ refer to the Mathematica built-in operations IntegerQ and Plus (if we implement THEOREM∀ within Mathematica). Now we can define, for example, the following simple *tower of domains*:

$$\begin{aligned}&\mathbb{Z} := \text{IntegerNumbers}[]\\ &\mathbb{Z}2 := \text{DirectProduct}[\mathbb{Z}]\\ &\mathbb{Z}22 := \text{DirectProduct}[\mathbb{Z}2]\end{aligned}$$

We can now evaluate arbitrary expressions over $\mathbb{Z}$, $\mathbb{Z}2$, and $\mathbb{Z}22$. For example, the terms

$$2 \underset{\mathbb{Z}}{+} 3, \underset{\mathbb{Z}}{\in}[3], \underset{\mathbb{Z}}{\in}[3.5], \langle 4, 7\rangle \underset{\mathbb{Z}2}{+} \langle 2, 14\rangle, \text{ and } \langle\langle 4, 7\rangle, \langle -5, 7\rangle\rangle \underset{\mathbb{Z}22}{+} \langle\langle 1, -2\rangle, \langle -3, 5\rangle\rangle$$

evaluate to 5, True, False, $\langle 6, 21\rangle$, and $\langle\langle 5, 5\rangle, \langle -8, 12\rangle\rangle$, respectively.

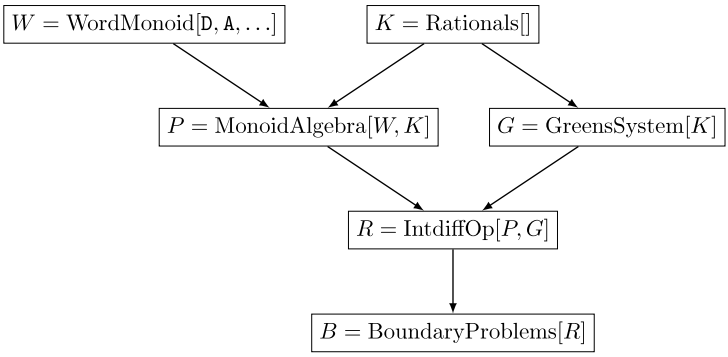

Fig. 1. Hierarchy of domains and functors

Let us also give an example of a simple *category* with a corresponding *conservation theorem* in the TH∃OREM∀ language:

$$\text{IsCommutid}[U] \Leftrightarrow \bigwedge \left\{ \begin{array}{l} \forall\limits_{\substack{x \in U[x], \\ y \in U[y]}} x +\!\!+_U y \in U[x +\!\!+_U y] \\ \forall\limits_{\substack{x \in U[x], \\ y \in U[y]}} x +\!\!+_U y = y +\!\!+_U x \end{array} \right.$$

$$\text{IsCommutid}[U] \Rightarrow \text{IsCommutid}[\text{DirectProduct}[U]]$$

In words: The domain U is a *commutid* (a commutative magma) with respect to the operation $+$ in U iff, for all objects x and y belonging to U, also $U[+][x, y]$ belongs to U and is equal to $U[+][y, x]$.

It is the ultimate goal of TH∃OREM∀ to support the *automated proof* of such theorems (as any other theorems). Of course, the difficulty of such proofs depends on the semantic complexity of the formulas in the functors and category descriptions. Typically, the operations in algorithmic functors are defined by induction, and the properties in algebraic categories are expressed by equalities. However, in a general build-up of mathematics by functors and categories, the definitions of operations in category description may be arbitrary formulas. Hence one must expect that the (automated) proof of conservation theorems can be arbitrarily complex. In fact, in the PhD thesis (Tomuţa, 1998), the proof of simple conservation theorems for algebraic categories and simple functors were already generated automatically within TH∃OREM∀. The conservation theorems in the hierarchy of algebraic domains needed in our algorithm for linear boundary problems are still proved by hand.

Let us now sketch the *hierarchy of domains and functors* we build up for the symbolic treatment of linear boundary problems (Fig. 1). As indicated before, the core piece of the symbolic engine is the algorithmic treatment of the relevant operators—the algebra $\mathcal{F}[\partial, \int]$ of integro-differential operators (Section 5). It is created as a domain R by the functor Intdiff-Op, based on two input domains P and G. The domain R is needed for various operations on boundary problems like solving, multiplying and factoring (see the following sections); these operations are bundled in the domain B created from R by the functor Boundary-Problems.

The domain R is constructed from P and G. Roughly speaking, P is the *free algebra* (noncommutative polynomials) in certain indeterminates while G is an *ideal of relations*. In this simplified picture, we can think of R as the quotient algebra P/G; using a Groebner basis for G, this boils down to reducing modulo G. The free algebra P is created by the functor MonoidAlgebra, which takes as input domains a term monoid W and a coefficient field K. The latter is created by the introduction functor Rationals, the former by the functor WordMonoid that creates the words over the specified letters: here the basic operators D for differentiation, A for integration ("antiderivative"), and other operators for multiplying by basis functions and extracting boundary values. The functor Green-System creates the relations between D, A, … over the ground field K, the so-called Green's system of interactions encoding the integro-differential axioms on the operator level (see Rosenkranz and Regensburger, 2008b, Section 3 for a detailed description).

The symbolic theory of boundary problems can be *organized* into the following three blocks treated in Sections 3–5, respectively: (1) Integro-differential algebras are the fundamental algebraic structure on which to base all subsequent operator algebras. (2) The purely linear aspects of boundary problems and Green's operators are best treated in isolation of any integro-differential structure. (3) Finally, one puts the two previous blocks together for expressing, solving and factoring boundary problems in their suitable operator algebras.

3. Integro-differential algebras

As we have said above, solving a linear boundary problem really means finding its Green's operator – an integral operator mapping forcing functions to solutions (see the next two sections for details). Hence the most important task for handling boundary problems symbolically is to master *integration*: to incorporate it into the algebraic structure and to relate it properly to its antipode, the derivation. This was achieved for the first time in Rosenkranz (2003, Def 11), Rosenkranz (2005, Def 2) by creating the notion of "analytic algebra".

We have subsequently reformulated this notion in a differential algebra context (Rosenkranz and Regensburger, 2008b, Def 4), now and henceforth using the name *integro-differential algebra* for avoiding confusion (since analytic algebras are commonly understood as homomorphic images of power series algebras). In this new language, our former "analytic algebras" appear as two dually paired integro-differential algebras (Rosenkranz and Regensburger, 2008b, Ex 10)". Simultaneous to but independent of (Rosenkranz and Regensburger, 2008b), the similar notion of differential Rota–Baxter algebras was introduced in Guo and Keigher (2008); see below for a short discussion of the relations.

Definition 1. We call $(\mathcal{F}, \partial, \int)$ an *integro-differential algebra* if $\mathcal{F}$ is a commutative algebra over a field K with K-linear operations ∂ and $\int$ such that

$$(\textstyle\int f)' = f, \tag{1}$$
$$(fg)' = f'g + fg', \tag{2}$$
$$(\textstyle\int f')(\int g') + \int (fg)' = (\int f')g + f(\int g') \tag{3}$$

holds, where $\ldots'$ is the usual shorthand notation for ∂. Note that we employ operator notation for ∂, $\int$ and the evaluation E introduced below.

This definition, taken from Regensburger and Rosenkranz (2009b), differs slightly from Rosenkranz and Regensburger (2008b) but turns out to be equivalent to it, as we shall see shortly. The axioms are motivated by the standard example $\mathcal{S} = C^\infty(\mathbb{R})$, an integro-differential algebra over $K = \mathbb{R}$ with derivation $\partial f = df/dx$ and integral $\int f = \int_0^x f(\xi)\, d\xi$. The *section axiom* (1) requires that $\int$ be a K-linear right inverse (= section) of ∂, in $\mathcal{S}$ provided by the fundamental theorem of calculus. While the *Leibniz axiom* (2) is the usual product rule of differentiation in $\mathcal{S}$, the *differential Baxter axiom* (3) is a version of the corresponding product rule of integration (see below for simpler formulations).

While $\int$ is, by definition, a right inverse of ∂, it is *never* a left inverse. In fact, the corresponding defect yields the *evaluation* $\mathrm{E} = 1 - \int\partial$ associated to $(\mathcal{F}, \partial, \int)$, which is crucial for specifying boundary conditions at a given point. It follows from the above axioms that E is a multiplicative projector, meaning E is idempotent: $\mathrm{E}(\mathrm{E}f) = \mathrm{E}f$ and multiplicative: $\mathrm{E}fg = (\mathrm{E}f)(\mathrm{E}g)$. In the standard example $\mathcal{S}$ we have of course $\mathrm{E}f = f(0)$. The existence of a nontrivial evaluation E is an important distinction to localizations of differential operator rings and pseudo-differential operators (Olver, 1993, p. 318). See Regensburger et al. (2009) for some investigations of the subtle relations between $\int$ and ∂^{-1}.

Since E is a projector, we obtain at once a direct decomposition $\mathcal{F} = \mathcal{C} + \mathcal{I}$ into the K-subspaces $\mathcal{C} = \mathrm{Im}(\mathrm{E})$ and $\mathcal{I} = \mathrm{Ker}(\mathrm{E})$. While the space $\mathcal{C}$ is the well-known *ring of constants* of the differential algebra $(\mathcal{F}, \partial)$, the space $\mathcal{I}$ is peculiar to integro-differential structures and turns out to be an ideal. In $\mathcal{S}$ it is clear that $\mathcal{C} = \mathbb{R}$ consists of the constant functions while $\mathcal{I}$ consists of the functions f initialized at zero in the sense that $f(0) = 0$. Hence we call $\mathcal{I}$ the *ideal of initialized functions*.

As one will perhaps expect from calculus, one can *rephrase the differential Baxter axiom* (3) in various simpler ways. The following proposition confirms this expectation.

Proposition 2. *In the presence of* (1) *and* (2)*, the differential Baxter axiom* (3) *is equivalent to each of the following conditions:*

$$\text{Integration by parts:}\quad f\int g = \int fg + \int f'\int g \tag{4}$$

$$\text{Evaluation variant:}\quad \int fg' = (fg)\big|_0^x - \int f'g \tag{5}$$

$$\text{Pure Baxter axiom:}\quad (\textstyle\int f)(\int g) = \int f\int g + \int g\int f \text{ and } \mathcal{C}\text{-linearity of } \int \tag{6}$$

$$\text{Multiplicativity of the evaluation:}\quad \mathrm{E}fg = (\mathrm{E}f)(\mathrm{E}g) \tag{7}$$

$$\text{Idealship of the initialized space:}\quad \mathcal{I} \trianglelefteq \mathcal{F} \tag{8}$$

Here $f\big|_0^x$ *is used as an abbreviation for* $f - \mathrm{E}f$*.*

It is an easy exercise to prove these equivalences. But it is perhaps not so obvious that the seemingly harmless *extra condition for the pure Baxter axiom* cannot be dropped; for a counterexample see Rosenkranz and Regensburger (2008b, Ex 7). From the perspective of analysis the extra condition is of course very natural since one should expect the integral to be linear not only over K but also over the constants $\mathcal{C}$.

Let us pause for a moment to reflect the consequences: The pure Baxter axiom is the only way of phrasing the product rule of integration without referring to ∂. One can therefore distill the pure integration structure $(\mathcal{F}, \int)$, known as Rota–Baxter algebra (Guo, 2002; Baxter, 1960; Rota, 1969), just as one usually studies differential algebras $(\mathcal{F}, \partial)$ without considering $\int\colon \mathcal{F} \to \mathcal{F}$. If the pure Baxter axiom were sufficient to ensure an integro-differential structure on $\mathcal{F}$, the only *coupling* between ∂ and $\int$ would be through the section axiom (1), so an integro-differential algebra $(\mathcal{F}, \partial, \int)$ would essentially split into the differential algebra structure $(\mathcal{F}, \partial)$ and the Baxter algebra structure $(\mathcal{F}, \int)$. In contrast, the differential Baxter axiom (3) requires a much tighter coupling between the differential and the Baxter structure of $\mathcal{F}$. This is the main difference to the differential Rota–Baxter algebras of Guo and Keigher (2008), where (1) is indeed the only interaction between ∂ and $\int$.

The reader may wonder why we have chosen the unwieldy axiom (3) rather than one of the simpler conditions in Proposition 2. One reason is that we want an identity similar to the Leibniz rule that works also in the *noncommutative case*. Without pursuing it any further, we mention that *e.g.* the $n \times n$ matrices with entries in $\mathcal{S}$ and componentwise ∂ and $\int$ are also an integro-differential algebra in the sense of Definition 1, except for the lack of commutativity (and with a ring instead of a field K). For characterizing such noncommutative integro-differential algebras one must double the identities given above: left and a right integration by parts, left and right evaluation variant, pure Baxter axiom with $\int$ being left and right $\mathcal{C}$-linear.

Another reason why (3) is the more powerful axiom in a general context is that it stays unchanged in the case of *nonzero weights*. Again we will not need this case here but it is good to know for seeing the bigger picture. For discrete models like those when ∂ is a difference operator and $\int$ a summation operator, one has to introduce weight terms $\lambda\,(\partial f)(\partial g)$ on the right-hand side of (2) and $\lambda\int fg$ on the right-hand side of the pure Baxter axiom (Guo and Keigher, 2008). In contrast, the differential Baxter axiom (3) remains unchanged.

Note that we have required K to be a *field* but most of the theory works for rings as well (or else would be interesting to adapt). In the sequel, however, we will only need fields. In fact, we will also require K to be of *characteristic zero*. The only nontrivial example of an integro-differential algebra in positive characteristic, as far as we are aware, is the ring of Hurwitz series (Keigher, 1997); see Rosenkranz and Regensburger (2008b, Ex 21) for a short description.

Our standard example $\mathcal{S} = C^\infty(\mathbb{R})$ is good for illustrative purposes but of course useless for serious algorithmic tasks. We will shortly give some more symbolic examples. But before this let us mention a few more *analytic examples*. Of course one may take the smooth function $C^\infty(U)$ or the analytic functions $C^\omega(U)$ on any open set $U \subseteq \mathbb{R}$; in this case ∂ has the usual meaning and the integral is $\int_a^x$ for an arbitrary initialization point $a \in U$. The same works for $U \subseteq \mathbb{C}$ if it is simply connected and one takes $C^\omega(U)$ to be the holomorphic functions on U. Note that the latter contains the Hardy spaces H^p $(p \geq 1)$ if U is the open unit disk (Yosida, 1995). They form a subspace of L^p, for $p = 2$ even a reproducing kernel Hilbert space. But of course H^p is not an algebra (let alone an integro-differential one).

Turning to the *algebraic examples*, the prototypical case is of course the ring of polynomials $K[x]$ with derivation and integral defined formally by setting $\partial x^n = n x^{n-1}$ and $\int x^n = x^{n+1}/(n+1)$. It is prototypical in a deeper sense: In fact, every integro-differential algebra contains a copy of $K[x]$, just as every field of characteristic zero contains a copy of the prime field $\mathbb{Q}$. The following proposition is taken from Regensburger and Rosenkranz (2009b), where the (simple) proof can also be found.

Proposition 3. *Let $(\mathcal{F}, \partial, \int)$ be an integro-differential algebra over a field K of characteristic zero. Then $x \mapsto \int 1$ induces a monomorphism $K[x] \to \mathcal{F}$ in the category of integro-differential algebras.*

Starting from $K[x]$, one may *adjoin other functions*; the easiest example is probably the exponential polynomials where one adds $e^{\lambda x}$ with $\partial e^{\lambda x} = \lambda e^{\lambda x}$ and $\int e^{\lambda x} = (e^{\lambda x} - 1)/\lambda$ for $\lambda \neq 0$. Here λ ranges over an additively closed set $\Lambda \subseteq K$, typical choices being $\pm\mathbb{N}$ and $\mathbb{Z}$ and $\mathbb{Q}$; in Albrecher et al. (2010) we have used the cases $\pm\mathbb{R} + \mathbb{R}i$ for modeling "stable" and "unstable" functions in an insurance context. By a straightforward calculation one may verify that the integral defined above leads to the explicit formula

$$\int x^k e^{\lambda x} = \tfrac{(-1)^{k+1}k!}{\lambda^{k+1}} + \sum_{i=0}^{k} \tfrac{(-1)^i k^{\underline{i}}}{\lambda^{i+1}} x^{k-i} e^{\lambda x} \ (\lambda \neq 0), \quad \int x^k = \tfrac{x^{k+1}}{k+1}$$

for specifying $\int$ on the K-basis $x^k e^{\lambda x}$.

The example of exponential polynomials already points to a complication that distinguishes integro-differential algebras from differential ones: In the latter case, the Leibniz axiom ensures that the derivation is fixed by prescribing it on the generators (the "functions" being adjoined), so closure under ∂ is automatic. Closure under $\int$ is not so easy to achieve. Suppose we adjoin x^{-1} to $K[x]$, obtaining the *Laurent polynomials* $K[x, x^{-1}]$. Since we need an antiderivative for x^{-1}, we are then forced to adjoin $\log x$ as well. Now we must check that every element in $K[x, x^{-1}, \log x]$ has an antiderivative. In this case the answer is yes, and one can derive the explicit formula

$$\int x^{-1} \log^n x = \tfrac{1}{n+1} \log^{n+1} x,$$

$$\int x^m \log^n x = \tfrac{(-1)^{n+1} n!}{(m+1)^{n+1}} + \sum_{k=0}^{n} \tfrac{(-1)^k n^{\underline{k}}}{(m+1)^{k+1}} x^{m+1} \log^{n-k} x \quad (m \neq -1),$$

where $n \in \mathbb{N}$, $m \in \mathbb{Z}$, and $n^{\underline{k}}$ denotes the falling factorial. Since $\log x$ is singular at $x = 0$, we use the integral $\int = \int_1^x$ here (understood in the algebraic way). We conclude that $K[x, x^{-1}, \log x]$ is an integro-differential algebra.

One sees from the previous example that, in general, an adjunction may lead to an indefinite chain of new adjunctions enforced by closure under $\int$. In this connection it would be interesting to investigate the following two problems:

- *Integro-Differential Completion:* Given a differential algebra $(\mathcal{F}, \partial)$, construct an integro-differential algebra $(\tilde{\mathcal{F}}, \partial, \int)$ such that $(\mathcal{F}, \partial) \leq (\tilde{\mathcal{F}}, \partial)$ is an extension of differential algebras.
- *Integro-Differential Closure:* Given an integro-differential algebra $(\mathcal{F}, \partial, \int)$, construct an extension $(\tilde{\mathcal{F}}, \partial, \int)$ such that every monic linear differential equation with coefficients from $\tilde{\mathcal{F}}$ has a solution in $\tilde{\mathcal{F}}$.

In the terminology of Rosenkranz and Regensburger (2008b, Def 18), the latter means that $\tilde{\mathcal{F}}$ is saturated and that we can also solve arbitrary boundary problems over $\tilde{\mathcal{F}}$.

Since these extension problems are very similar in spirit to the differential field extension "towers" used in symbolic integration (Bronstein, 2005) and differential Galois theory (van der Put and Singer, 2003), a few words of clarification are necessary. Firstly it is clear that a field cannot carry an integro-differential structure since $\mathcal{I} = \operatorname{Ker}(\mathrm{E})$ is always a nontrivial proper ideal. Secondly, one must distinguish *an antiderivative* F of an element $f \in \mathcal{F}$ from the well-defined element $\int f$. Here the point is that the differential Baxter axiom enforces a consistent choice of the integration constants, across all of $\mathcal{F}$; this is also crucial for the computation of Green's operators. This leads to an intriguing and important problem: How can we use the well-developed theory of differential field extensions for constructing and understanding integro-differential extensions?

As an alternative to the approach via adjunction, one may also employ *holonomic functions* since they are closed under differentiation and definite integration (Chyzak, 1994; Salvy and Zimmerman, 1994). In other words, they form a constructive integro-differential subalgebra of the integro-differential algebra $K[[x]]$ of formal power series.

Up to now we have only mentioned *ordinary integro-differential algebras* in the sense of Rosenkranz and Regensburger (2008b, Def 8), meaning $\mathcal{C} = \mathrm{Ker}(\partial) = K$. In this case, the extra condition in (6) is of course redundant, and E is actually a multiplicative linear functional, *i.e.* a character on $\mathcal{F}$. When dealing with boundary problems for partial differential equations, however, $\mathcal{C}$ will typically be infinite-dimensional. The simplest example is $K[x, y]$ with $\partial = d/dx$ and hence $\mathcal{C} = K[y]$; for a slightly more complicated example see Rosenkranz and Regensburger (2008b, Ex 7).

In such a case one clearly needs some *more structure*: at least several derivations $\partial_{x_1}, \ldots, \partial_{x_n}$ and several integrals $\int^{x_1}, \ldots, \int^{x_n}$. Of course each $(\partial_{x_i}, \int^{x_i})$ must form an integro-differential structure, but we must also know something more about the interaction among the derivations and among the integrals. One possibility is to require a right action $\odot$ of the multiplicative monoid $K^{n\times n}$ on $\mathcal{F}$ such that relations like

$$\partial_{x_j}(f \odot M) = \sum_{i=1}^{n} M_{ji}(\partial_{x_i} f) \odot M$$

are satisfied for all $f \in \mathcal{F}$ and $M \in K^{n\times n}$. Of course this is just the chain rule, applied to a linear change of coordinates M. The corresponding relations for the integral (substitution rule) are more complicated. An approach along these lines is sketched in Rosenkranz et al. (2009) for the special case $n = 2$ and $\mathcal{T} = C^\infty(\mathbb{R} \times \mathbb{R})$. It is essential that we allow all of $K^{n\times n}$ and not only $\mathrm{GL}(n, K)$ since projectors are needed for constructing Green's operators. We plan to present this construction in a subsequent paper.

From Proposition 3 we know that $K[x_i]$ is always contained in $\mathcal{F}$, so it is tempting to extend this approach to include *polynomial substitutions* instead of just linear ones, with a right action of the monoid $(K[x_1, \ldots, x_n]^n, \circ)$ instead of $K^{n\times n}$. Most likely this will lead to rather complicated relations. Here we are just at the beginning of a fascinating new field of research: As also observed in Plesken and Robertz (2010, p. 232), a systematic treatment of functional composition and the chain rule seems to be lacking in differential algebra.

4. Abstract boundary problems

A *boundary problem* consists of a differential equation and boundary conditions (as mentioned before: everything is linear). Based on some integro-differential algebra $\mathcal{F}$, the differential equation can be expressed as $Tu = f$ with $T\colon \mathcal{F} \to \mathcal{F}$. Here f is to be regarded as a symbolic parameter, so that we search the solution u in terms of a "generic" function f. Since differential operators are far from injective, we need boundary conditions for ensuring that we search *the* solution u in terms of f. They are given as $\beta_i(u) = 0$ $(i \in I)$, where $\beta_i \in \mathcal{F}^*$ are linear functionals. Summarizing, we express a boundary problem as follows: Given $f \in \mathcal{F}$, find $u \in \mathcal{F}$ such that

$$\boxed{\begin{array}{l} Tu = f, \\ \beta_i u = 0 \ (i \in I) \end{array}} \tag{9}$$

is satisfied. In this context, f is usually called the forcing function.

As suggested before, we expect the β_i to be such that we have a *regular* boundary problem in the sense that for each forcing function $f \in \mathcal{F}$ there is exactly one $u \in \mathcal{F}$ such that (9) is satisfied. The symbolic treatment of singular boundary problems is a highly interesting extension topic (Korporal et al., 2011), but here it would lead us too far afield. Note also that we have restricted ourselves to so-called semi-inhomogeneous boundary problems (*i.e.* the differential equation is inhomogeneous but the boundary conditions are homogeneous). For ordinary differential equations this is no essential limitation (Rosenkranz and Regensburger, 2008b, p. 516), but for partial differential equations a thorough investigation will be needed: By Duhamel's principle (Renardy and Rogers, 2004,

Exc 1.23) one can, to some extent, shift inhomogeneities between differential equation and boundary conditions.

For an ordinary differential equation of order n, we need n boundary conditions so $I = \{1, \ldots, n\}$ in that case. For a *partial differential equation*, things are more involved. Here one must specify u or some of its derivatives on suitable submanifolds; typical choices are Dirichlet or Neumann conditions. In general it can be difficult, even for linear problems, to determine exactly where to impose which condition. For that purpose, it can be of advantage to apply involutive completion (Gerdt, 1999). These issues are often avoided by assuming the given partial differential equation(s) in Cauchy–Kovalevskaya form, where the situation is clear (Olver, 1993, Thm 2.73). For our present purposes we assume suitable boundary conditions, howsoever they are found. The crucial point here is that they can always be expressed in the form $\beta_i(u) = 0$ $(i \in I)$ with $\beta_i \in \mathcal{F}^*$, no matter whether we impose Dirichlet or Neumann conditions, combined or other conditions. For example, the Neumann condition $\partial_y u(x, 0) = 0$ for $u \in C^\infty(\mathbb{R}^2)$ would have $I = \mathbb{R}$ and $\beta_x(u) = \partial_y u(x, 0)$. Note that we allow also global conditions like $\beta_x(u) = \int_0^1 u(x, y)\, dy$; this will be important later (Example 15).

The collection β_i $(i \in I)$ is not an invariant way of imposing the boundary conditions. Since they are linear, whenever $\beta(u) = 0$ and $\tilde{\beta}(u) = 0$ we have also $(c\beta + \tilde{c}\tilde{\beta})(u) = 0$ for any $c, \tilde{c} \in K$. This means the invariant object described by the boundary conditions β_i $(i \in I)$ is the *boundary space*

$$\mathcal{B} = [\beta_i \mid i \in I] \leq \mathcal{F}^*$$

spanned by the β_i over K. To be precise, we would have to take the so-called orthogonal closure (Regensburger and Rosenkranz, 2009a, A.1); we omit these technical details here for focusing on an intuitive overview. Suffice it to say that boundary spaces have a role similar to that of (radical) ideals in algebraic geometry (Rosenkranz and Regensburger, 2008b, p. 531). At this point our treatment of boundary conditions diverges radically from (functional) analysis where a condition like $\partial_y u(x, 0) = 0$ would be understood in terms of the operator $u(x, y) \mapsto \partial_y u(x, 0)$, usually perceived as a trace operator between suitable Sobolev spaces. The advantage of the space approach will become apparent when defining the composition of boundary problems (Definition (13)).

Looking back to the prototypical boundary problem (9), we can now combine the necessary data into the pair $(T, \mathcal{B})$. It is now just a small step to the notion of an *abstract boundary problem*: We strip $\mathcal{F}$ from its integro-differential structure, viewing it as a naked (though generally infinite-dimensional) K-vector space. In that case $\mathcal{B} \leq \mathcal{F}^*$ still makes sense as an arbitrary subspace, but we must amend our requirement that T be a differential operator. All we require from T in the abstract case is that it be surjective. Of course this corresponds to a strong assertion in analysis: In the standard example 8, this says that every inhomogeneous differential equation $Tu = f$ with a monic C^∞ differential operator T has a C^∞ solution.

Definition 4. A *abstract boundary problem* on a K-vector space $\mathcal{F}$ is a pair $(T, \mathcal{B})$ with an epimorphism $T\colon \mathcal{F} \to \mathcal{F}$ and an orthogonally closed subspace $\mathcal{B} \leq \mathcal{F}^*$. It is called *regular* if for each $f \in \mathcal{F}$ there is a unique $u \in \mathcal{F}$ with $Tu = f$ and $u \in \mathcal{B}^\perp$.

As mentioned before, the property of a subspace $\mathcal{B} \leq \mathcal{F}^*$ to be orthogonally closed is of a technical nature, and we will not enter into the details here. It is not necessary for ordinary differential equations (finite-dimensional boundary spaces are always orthogonally closed), but it is inevitable for partial differential equations. The point is that a condition like $\partial_y u(x, 0) = 0$ is not exhausted by the span of the individual functionals $\beta_x(u) = \partial_y u(x, 0)$ since we can create many other conditions, e.g. integrals like $\partial_y \int_3^5 u(\xi, 0)\, d\xi$. All of these are included in the *orthogonal closure* of the β_x, the smallest orthogonally closed subspace containing the β_x.

Abstract boundary problems have not been introduced for the sake of abstraction but because they encompass at once a range of *special cases*, in particular linear ordinary and partial differential equations and linear ordinary and partial differential systems (and probably also certain discrete and functional systems though we have not yet investigated this case). The point is that various operations and properties of boundary problems can be defined and investigated most economically in pure linear algebra.

For example, one verifies at once that $(T, \mathcal{B})$ is regular iff $\mathcal{F} = \operatorname{Ker}(T) \dotplus \mathcal{B}^\perp$, where $\mathcal{B}^\perp \leq \mathcal{F}$ is defined as $\{u \in \mathcal{F} \mid \beta(u) = 0 \text{ for all } \beta \in \mathcal{B}\}$. If both $\operatorname{Ker}(T)$ and $\mathcal{B}$ are n-dimensional, say

with $\mathrm{Ker}(T) = [u_1, \ldots, u_n]$ and $\mathcal{B} = [\beta_1, \ldots, \beta_n]$, then the regularity of $(T, \mathcal{B})$ is equivalent to the regularity of the evaluation matrix $[\beta_j(u_i)] \in K^{n\times n}$. This applies to the case of boundary problems for ordinary differential equations. In any case, we define the *Green's operator* of an abstract boundary problem $(T, \mathcal{B})$ as the operator $G\colon \mathcal{F} \to \mathcal{F}$ that assigns to each $f \in \mathcal{F}$ the unique $u \in \mathcal{F}$ with $Tu = f$ and $u \in \mathcal{B}^\perp$. In other words, G is characterized by $TG = 1_{\mathcal{F}}$ and $\mathrm{Im}(G) = \mathcal{B}^\perp$. For reasons that will become clear soon, we write $(T, \mathcal{B})^{-1}$ for G.

Example 5. In the sequel, we will illustrate the abstract notions of this chapter in the following *running example*, the simplest possible (true) boundary problem for a LODE; see Section 3.2 in Rosenkranz (2005). In traditional notation, we require for a given forcing function $f \in C^\infty[0, 1]$ a function $u \in C^\infty[0, 1]$ such that

$$\boxed{\begin{aligned} &u'' = f, \\ &u(0) = u(1) = 0. \end{aligned}} \tag{10}$$

Here we work with the real vector space $\mathcal{F} = C^\infty[0, 1]$, the differential operator $T = D^2$, and the boundary space $\mathcal{B} = [L, R] \leq \mathcal{F}^*$ spanned by the functionals $Lu = u(0)$ and $Ru = u(1)$ of evaluation at the left and right boundary point, respectively. As mentioned above, $\mathcal{B}$ is orthogonally closed since it is a finite-dimensional subspace of $\mathcal{F}$. Furthermore, one may immediately confirm (by elementary methods) that this boundary problem is regular: For every $f \in C^\infty[0, 1]$, there is exactly one $u \in \mathbb{C}^\infty[0, 1]$ such that (10) is satisfied.

Example 6. As a simple LPDE example, consider the following (inhomogeneous) version of a wave equation (usually called one-dimensional since the time coordinate is not counted), taken from Section 7 in Regensburger and Rosenkranz (2009a). Given $f \in C^\omega(\mathbb{R}^2)$, we want to find $u \in C^\omega(\mathbb{R}^2)$ such that

$$\boxed{\begin{aligned} &u_{tt} - u_{xx} = f \\ &u(x, 0) = u(x, x) = 0. \end{aligned}} \tag{11}$$

The boundary conditions effectively prescribe two wave nodes (one stationary at the origin, the other one moving at unit speed). In this example, we use as the underlying vector space $\mathcal{F} = C^\omega(\mathbb{R}^2)$, the real-analytic functions. The boundary space $\mathcal{B}$ must be defined as the orthogonal closure of the span (within $\mathcal{F}^*$) of the infinite families of functionals $\iota_x u(x, t) = u(x, 0)$ and $\kappa_x u(x, t) = u(x, x)$, where x ranges over $\mathbb{R}$. The differential operator of this problem will be written as $T = \partial_{tt} - \partial_{xx}$. One may check that (11) is regular by using a power-series ansatz and confirming that the coefficients of $u(x, t)$ are determined uniquely in terms of the coefficients of $f(x, t)$.

For ordinary differential equations one distinguishes boundary problems from *initial value problems*. The reason is that the latter are generally much more tractable, both theoretically and numerically. The same turns out to be true from a symbolic perspective: An n-th order differential equation $Tu = f$ is easier to solve if one imposes the initial conditions $u(0) = u'(0) = \cdots = u^{n-1}(0) = 0$ rather than general boundary conditions $\beta_1(u) = \cdots \beta_n(u) = 0$. Hence it is useful to split the computation of a Green's operator for (9) in two parts: First we compute the simpler Green's operator $\tilde{G}$ for the associated initial value problem (with boundary conditions replaced by initial conditions), then we "rotate" the operator $\tilde{G}$ into the actual Green's operator G satisfying the given boundary conditions. This strategy generalizes to abstract boundary problems.

Proposition 7. *Given any section $\tilde{G}$ of T, the Green's operator $G = (T, \mathcal{B})^{-1}$ can be computed by*

$$G = (1 - P)\tilde{G},$$

where P is the projector with $\mathrm{Im}(P) = \mathrm{Ker}(T)$ and $\mathrm{Ker}(P) = \mathcal{B}^\perp$.

Of course this works only in as much as we can handle *projectors* effectively. In the case of ordinary differential equations (where $\dim \mathcal{B} = \dim \mathrm{Ker}(T) < \infty$), it is clear how to proceed (Regensburger and Rosenkranz, 2009a, Section 6), the case of partial differential equations is far more complicated and will be described in subsequent papers.

Example 8. Going back to the boundary problem (10), it is clear that one can easily obtain a section $\tilde{G}$ of $T = D^2$ by just integrating twice. So if A denotes the usual antiderivative operator

$$Au(x) = \int_0^x u(\xi)\, d\xi,$$

we can define $\tilde{G} = A^2$. The projector P can be determined by a simple linear interpolation: Since its image is required to be $\mathrm{Ker}(T) = [1, x]$, we can use the ansatz $Pu(x) = \alpha(u) + \beta(u)\, x$ for $\alpha(u), \beta(u) \in \mathbb{R}$ to be determined in dependence on u. This can be done by using the condition $\mathrm{Ker}(P) = \mathrm{Im}(1 - P) = \mathcal{B}^\perp$, which implies that $(1 - P)u = u - Pu$ satisfies the given boundary conditions $u(0) = u(1) = 0$ for any function u. In other words, we must require that Pu coincide with u on the interval endpoints 0 and 1. This yields immediately $\alpha(u) = u(0)$ and $\beta(u) = u(1) - u(0)$, so we can write the projector in operator language as $P = L + x\,(R - L)$. As detailed in Rosenkranz (2005), one immediately computes the Green's operator G from P and $\tilde{G} = A^2$. Moreover, there is a systematic procedure to extract the Green's function corresponding to a Green's operator (Rosenkranz and Regensburger, 2008b).

Example 9. Let us now look at the LPDE example (11). In that case, $\mathrm{Ker}(T)$ is just the solution space of the homogeneous wave equation, comprising all functions of the form

$$\frac{1}{2}\Big(g(x-t) + g(x+t)\Big) + \frac{1}{2}\int_{x-t}^{x+t} h(y)\, dy \tag{12}$$

according to the well-known d'Alembert formula (Evans, 1998, p. 68). Again we have $\mathrm{Im}(P) = \mathrm{Ker}(T)$, so the projector P maps a given function $u \in C^\infty(\mathbb{R}^2)$ to a function (12) with certain function $g, h \in C^\infty(\mathbb{R})$ depending on u. So the situation is rather similar to the previous example except that here P depends on two univariate *functions* instead of two *numbers* α, β. In analogy to the LODE example, $\mathrm{Ker}(P) = \mathrm{Im}(1 - P) = \mathcal{B}^\perp$ now leads to the condition that $u(x, t)$ and $Pu(x, t)$ must coincide for $t = 0$ and $t = x$. This allows to determine g and h in terms of u. The first condition yields immediately $g(x) = u(x, 0)$ while the second can be transformed into

$$\frac{1}{2}\int_{x-t}^{x+t} h(y)\, dy = u\left(\tfrac{x+t}{2}, \tfrac{x+t}{2}\right) - u\left(\tfrac{x-t}{2}, \tfrac{x-t}{2}\right) + \frac{1}{2}\Big(u(x-t, 0) - u(x+t, 0)\Big).$$

Hence we end up with $Pu(x, t) = u(\frac{x+t}{2}, \frac{x+t}{2}) - u(\frac{x-t}{2}, \frac{x-t}{2}) + u(x - t, 0)$ or in operator notation

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}^* - \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}^* + \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix}^*,$$

using $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)^*$ for the substitution $f(x, t) \mapsto f(ax + bt, cx + dt)$. This involves now the monoid action of $K^{2\times 2}$ mentioned earlier; see also the comments given below after Example 14 and the more detailed treatment in Rosenkranz et al. (2009). From P and the section $\tilde{G}$ given by the d'Alembert formula (12), it is straightforward to compute the Green's operator G of (10).

One cannot expect to solve "every" boundary problem (except if one restricts to differential operators with constant coefficients). Hence it is advisable to adopt the old principle *divide et impera*: One tries to split off (symbolically) simpler problems that can be solved exactly and then one puts the resulting Green's operators together. This works even if some of those smaller problems cannot be done symbolically: In that case the corresponding Green's operator is a numerical simulation (and of course the overall result is of a hybrid nature). In which sense do we split boundary problems?

Definition 10. The *product* of boundary problems is defined by

$$(T, \mathcal{B}) \cdot (\tilde{T}, \tilde{\mathcal{B}}) = (T\tilde{T}, \tilde{T}^*(\mathcal{B}) + \tilde{\mathcal{B}}),$$

where $\tilde{T}^*$ denotes the adjoint of the operator $\tilde{T}$.

This multiplication is the *semidirect product* of operator composition ("iterated differentiation") and vector space addition ("adjoining boundary conditions"). If $\mathcal{B}$ is generated by β_i, the space $\tilde{T}^*(\mathcal{B})$ is generated by $\beta_i \circ \tilde{T}$, so the boundary conditions of the left problem are "lifted" by the differential operator of the right problem (if T and $\tilde{T}$ have respectively orders n and $\tilde{n}$ then the lifted boundary conditions will have order $n + \tilde{n}$).

Of course the whole point of this multiplication is that it allows us to compute Green's operators in a piecemeal fashion (Regensburger and Rosenkranz, 2009a, Prop 3.2). This means two things: First of all, we can be sure that the product problem is regular if the two constituent problems are. Hence the *Green's operator of the product problem* is well-defined, and it is given by

$$((T, \mathcal{B}) \cdot (\tilde{T}, \tilde{\mathcal{B}}))^{-1} = (\tilde{T}, \tilde{\mathcal{B}})^{-1} \cdot (T, \mathcal{B})^{-1}, \tag{13}$$

explaining the notation $(\ldots)^{-1}$ for the Green's operator of a boundary problem.

As pointed out before, the real problem is not to multiply given boundary problems but rather to *factor* a large problem into smaller ones. It is a very pleasing result that this is always possible—of course, provided we can factor the underlying differential operator (Regensburger and Rosenkranz, 2009a, Thm 4.8).

Theorem 11. *Let $(T, \mathcal{B})$ be a regular boundary problem. If $T = T_1T_2$ is a factorization into epimorphisms and $\mathcal{B}_2$ any subspace of $\mathcal{B}$ such that $(T_2, \mathcal{B}_2)$ is a regular boundary problem, then there is a unique left factor $(T_1, \mathcal{B}_1)$ for the factorization*

$$(T, \mathcal{B}) = (T_1, \mathcal{B}_1) \cdot (T_2, \mathcal{B}_2), \tag{14}$$

namely $\mathcal{B}_1 = \tilde{G}_2^*(\mathcal{F} \cap \mathrm{Ker}(T_2)^\perp)$ with $\tilde{G}_2$ an arbitrary section of T_2.

Practically speaking this means the following: Once we have chosen a right factor T_2 of T, we have a lot of freedom in choosing which boundary conditions from $\mathcal{B}$ we combine with it (as long as the resulting problem is regular—this is the generic case). But whatever conditions we have chosen for T_2, those for T_1 will always be the same: As one sees from the formula above, the boundary space $\mathcal{B}_2$ depends only on T_2 and not on $\mathcal{B}_1$. The fact that we can choose *any* section $\tilde{G}_2$ of T_2 is crucial since then we can get away with solving initial value problems (see the remarks above). But in a *cascade integration* one anyway wants to compute Green's operators: Splitting off a simple factor T_2, we are hopefully able to compute the full Green's operator $G_2 = (T_2, \mathcal{B}_2)^{-1}$. In that case, the determination of the left factor in (14) simplifies to $\mathcal{B}_1 = G_2^*(\mathcal{F})$. This means we obtain the corresponding boundary conditions immediately from G_2, without further elimination (computing the intersection). We are then left to compute the Green's operator $G_1 = (T_1, \mathcal{B}_2)^{-1}$, which might well be done numerically.

Example 12. For a simple factorization example, let us take up the boundary problem (10). Using our methods (Rosenkranz and Regensburger, 2008a, Ex. 28), this boundary problem can be factored as $(D^2, [L, R]) = (D, [F]) \cdot (D, [L])$, where $Fu = \int_0^1 u(\xi)\, d\xi$ denotes the operator of definite integration. In traditional notation, this factorization reads as follows:

$$\boxed{\begin{array}{l} u'' = f \\ u(0) = u(1) = 0 \end{array}} = \boxed{\begin{array}{l} u' = f \\ \int_0^1 u(\xi)\, d\xi = 0 \end{array}} \cdot \boxed{\begin{array}{l} u' = f \\ u(0) = 0 \end{array}}$$

Simple as it may be, this example nevertheless show already one characteristic feature of such factorizations: The left-hand factor typically involves a global condition. From our remarks above, it is clear that such a global condition cannot be avoided. But this is also clear on intuitive grounds: Otherwise we could reduce a boundary problem to a cascade of (first order) initial value problems, which runs counter to the intuition that one must somehow "connect" the two boundary points (experts may think of shooting methods). In our factorization, the connection is made by the F operator. In contrast, if one factors the initial value problem $(D^2, [LD, L])$, both factor problems come out as $(D, [L])$, and no global condition is needed for "connecting"; on the side of Green's operators, this factorization corresponds to the trivial factorization $A^2 = A \cdot A$.

Example 13. The factorization of (11) is only slightly more complicated. Using the methods of Regensburger and Rosenkranz (2009a), it can be factored as follows:

$$\boxed{\begin{array}{l} u_{tt} - u_{xx} = f \\ u(x,0) = u(x,x) = 0 \end{array}} = \boxed{\begin{array}{l} u_t - u_x = f \\ \int_0^x u(y,y)\,dy = 0 \end{array}} \cdot \boxed{\begin{array}{l} u_t + u_x = f \\ u(x,0) = 0 \end{array}}$$

Again the factorization $T = \partial_{tt} - \partial_{xx} = (\partial_t - \partial_x)(\partial_t + \partial_x)$ is straightforward, and we have a global condition in the right factor. But now the geometry is more interesting: As we are dealing with function u defined on the (x,t) plane, the global condition is now a line integral on the diagonal $x = t$. In this case, we can simplify the condition: Differentiating with respect to x, we regain the earlier boundary condition $u(x,x) = 0$ that stipulates that u vanish on the diagonal. So we can actually avoid the global condition, and the two boundary conditions of (11) are simply distributed to the factors $\partial_t - \partial_x$ and $\partial_t + \partial_x$ of T, supplying $u(x,x) = 0$ to the former and $u(x,0) = 0$ to the latter. Of course this is not always possible: In Example 15, we will see a case where the global condition in the left factor cannot be avoided.

The *abstract theory of boundary problems* can be developed much further, also beyond the results in Regensburger and Rosenkranz (2009a). There one can also find an interesting notion of dual boundary problems and a category structure on boundary problems. The setup used there allows boundary problems $(T, \mathcal{B})$ with epimorphisms $T\colon \mathcal{F}_1 \to \mathcal{F}_2$ between possibly different K-vector spaces $\mathcal{F}_1$ and $\mathcal{F}_2$. This is interesting for applications, where differentiation d/dx is often seen as a map $C^1(\mathbb{R}) \to C(\mathbb{R})$.

5. Algebraic boundary problems

The results in the preceding section are general, abstract—and inconstructive. For actually *computing* the Green's operator of a given boundary problem (9) or for factoring it into smaller problems, we need an algorithmic representation both for the operators T, G and for the boundary spaces $\mathcal{B} \leq \mathcal{F}^*$.

The key tool for writing down a Green's operator in an algorithmic representation is, essentially, *Groebner bases* (Buchberger, 1965, 1970, 1998), here in their noncommutative extension (Mora, 1986, 1994; Ufnarovski, 1998) since we want to write operators as polynomials (whose noncommutative multiplication is the composition of operators). The main idea is that we collect a few basic operators $S_1, S_2, \ldots$ like differentiation, integration and boundary functionals, and then describe their relations by a suitable ideal generated by a set of basic relations $R_1(S_1, S_2, \ldots) = 0$, $R_2(S_1, S_2, \ldots) = 0$. As is well-known, the normal form of an operator with respect to the basic relations $R_1, R_2, \ldots$ will only be unique if the latter form a (noncommutative) Groebner basis for the relation ideal.

For the case of *ordinary differential equations*, the detailed construction is presented in Rosenkranz and Regensburger (2008b, Section 3). In that case, one starts from an ordinary integro-differential algebra $(\mathcal{F}, \partial, \int)$ and creates the noncommutative polynomials in the indeterminates D for differentiation, A for integration ("antiderivative"), $f \in \mathcal{F}$ for all the corresponding multiplication operators (or a basis thereof), all or some characters $\varphi \in \mathcal{F}^*$ as needed for boundary conditions. Following the usual (but confusing) practice of renaming the indeterminates D and A into ∂ and $\int$, the resulting ring of ordinary integro-differential operators is denoted by $\mathcal{F}[\partial, \int]$.

The *operator relations* are essentially the following: Leibniz rule, Baxter rule, section rule, multiplicativity of characters; see Table 1 in Rosenkranz and Regensburger (2008b). They give rise to a noetherian and confluent rewrite system. For a thorough discussion of some of the rewriting aspects, see also Rosenkranz et al. (2011). At this point, let us just mention that the confluence proof can be done in two ways:

- Either one instantiates the multiplication operators f by all elements of an arbitrary (necessarily infinite) basis of $\mathcal{F}$ and appeals to the axioms of integro-differential algebras (in addition to the relations themselves) for showing that all S-polynomials vanish. This is the route taken in Rosenkranz and Regensburger (2008b).

- Or one employs the so-called integro-differential polynomials of Rosenkranz and Regensburger (2008a). They can be seen as nonlinear integral and differential operators, generalizing the differential polynomials (just as the latter describe a generic differential algebra extension, the former do the same for integro-differential algebras). Consequently, f is now taken as an indeterminate and hence a single element; this approach is described in Rosenkranz et al. (2011).

Both methods will establish the confluence of the system. Furthermore, the normal forms can be characterized explicitly.

Here is an *example* for a boundary problem for an ordinary differential equation, taken from the Appendix of Rosenkranz et al. (2009), that illustrates the difficulties encountered in current computer algebra systems when confronted with boundary problems of the kind considered here (*i.e.* with a symbolic right-hand side f).

Example 14. Let $\mathcal{F}$ be either the standard integro-differential algebra $\mathcal{I}$ or a suitable extension of $K[x]$ that contains e^x and e^{e^x}. Then consider the following third-order boundary problem over $\mathcal{F}$:

$$\boxed{\begin{aligned} &u''' - (e^x+2)u'' - u' + (e^x+2)u = f, \\ &u(0) = u(1) = u'(1) = 0. \end{aligned}} \tag{15}$$

The (rather lengthy) Green's operator of this boundary problem can be found in Rosenkranz et al. (2009). Experimenting with the boundary conditions a bit, one can observe that the support of computer algebra systems for boundary problems (with generic forcing functions) is somewhat ad hoc: For example, in the form above, neither Mathematica nor Maple was able to solve the problem when we published it (but both of them did solve the homogeneous differential equation). At the moment Mathematica (version 8) fails, as it cannot even solve the homogeneous differential equation. Interestingly, Maple (version 14) can solve it now, but slight perturbations of the boundary conditions will again lead to failure (for example adding the term $u''(0)$ to the third condition). This is all the more astonishing since the Green's operator does not need any adjunctions beyond those already needed for the homogeneous differential equations.

For *partial differential equations*, new complications enter the stage, and we regard the current setup as highly experimental (Rosenkranz et al., 2009); a more systematic treatment is in preparation. As pointed out in Section 3, partial integro-differential algebras are to be given somewhat more structure beyond the mere replication of $(\partial, \int)$ pairs. For the sake of simplicity, let us restrict to two derivations ∂_x, ∂_y with corresponding integrals $\int^x$, $\int^y$. They will be represented in the operator ring by indeterminates D_x, D_y and A_x, A_y, respectively. Of course, we will also adjoin indeterminates $f(x, y)$ representing multiplication operators for (some?) elements $f \in \mathcal{F}$; this includes at least the polynomials x^m and y^n. At this point, we do not take evaluations other than those for $x = 0$ and $y = 0$.

The really new objects correspond to the monoid action $\odot\colon K^{2\times 2} \times \mathcal{F} \to \mathcal{F}$. For reflecting this additional structure, we adjoin the matrix indeterminates

$$\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)^* \quad \text{with} \quad \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in K^{2\times 2}$$

denoting the substitutions $f(x, y) \mapsto f(ax+by, cx+dy)$. The asterisk is to remind of its contravariant nature, $(MN)^* = N^*M^*$ for all $M, N \in K^{2\times 2}$. The latter is one of the new relations in the resulting ring $\mathcal{F}[\partial_x, \partial_y, \int^x, \int^y]$ of *partial integro-differential operators*; other relations are needed for expressing certain instances of the chain rule and the substitution rule. Note that the substitutions for $x = 0$ and $y = 0$ are included as special substitutions.

Why have we included those *substitutions* in the partial operator ring? It turns out that even the simplest Green's operators simply cannot be expressed in terms of D_x, D_y, A_x, A_y and the multipliers alone. For example (writing t in place of y), the Green's operator of the unbounded inhomogeneous wave equation $(\partial_{xx} - \partial_{tt}, [T_x, T_x\partial_t])$, with the substitution $T_x u = u(x, 0)$, is given by an integral over the symmetric right-angled triangle extending downwards from $(x, t) \in \mathbb{R}^2$ up to the x-axis. We cannot express such an integration by the axis-parallel integrators A_x and A_y. See Rosenkranz et al. (2009) for the (simple) derivation of its Green's operator.

The inhomogeneous wave equation can also be restricted to the interval $x \in [0, 1]$, in which case it becomes a *boundary problem* in the narrower sense (while we tend to call any $\beta \in \mathcal{F}^*$ a boundary

condition, normal usage restricts them to evaluations taken uniformly at all times like $u(0,t) = 0$). The following example is taken from Regensburger and Rosenkranz (2009a, Section 7).

Example 15. Consider the boundary problem $\mathcal{P} = (\partial_{xx} - \partial_{tt}, [T_x, T_x\partial_t, L_t, R_t])$ with the left and right boundary conditions $L_t u = u(0,t)$ and $R_t u = u(1,t)$. Written in traditional notation, this means we want to find $u \in \mathcal{F}$

$$\boxed{\begin{aligned} &u_{xx} - u_{tt} = f, \\ &u(x,0) = u_t(x,0) = u(0,t) = u(1,t) = 0 \end{aligned}}$$

for a forcing function $f \in \mathcal{F}$. Here we use $\mathcal{F} = C^\infty(\mathbb{R}^2)$ although $C^\infty(\mathbb{R} \times \mathbb{R}^+)$ would be more appropriate from the viewpoint of analysis. Rather than writing down the Green's operator, let us factor this second-order problem into two first-order ones, using the obvious factorization $\partial_{xx} - \partial_{tt} = (\partial_t - \partial_x)(\partial_t + \partial_x)$ for the differential operator. For $\partial_t + \partial_x$ it is natural to take either $u(x,0) = u(0,t) = 0$ or $u(x,0) = u(1,t) = 0$ as conditions; let us take the former. Hence we split off the right factor $\mathcal{P}_2 = (\partial_t + \partial_x, [T_x, L_t])$. What is the boundary space of the left factor? It turns out to contain two consist of two parts: Firstly T_x again, which was to be expected since it is "lifted" to the $T_x\partial_t$ of $\mathcal{P}$ by the adjoint $(\partial_t + \partial_x)^*$. Secondly, it contains an integral along the characteristic line of $\partial_t + \partial_x$, namely

$$C_t u = \int_{\max(1-t,0)}^{1} u(\xi, \xi + t - 1)\, d\xi .$$

Altogether we obtain then $\mathcal{P}_1 = (\partial_t - \partial_x, [T_x, C_t])$. Written in traditional notation we have thus effected the factorization

$$\boxed{\begin{aligned} &u_t - u_x = f \\ &u(x,0) = \textstyle\int_{\max(1-t,0)}^{1} u(\xi, \xi + t - 1)\, d\xi = 0 \end{aligned}} \cdot \boxed{\begin{aligned} &u_t + u_x = f \\ &u(x,0) = u(0,t) = 0 \end{aligned}},$$

where the right-hand factor is an ordinary transport problem with an initial and boundary condition while the left-hand factor (which is unique by Theorem 11) contains a global condition.

As we have seen in Example 12, the appearance of *global conditions* in factor problems is not at all peculiar to partial differential equations. This is why the precise definition of $\mathcal{F}[\partial, \int]$ explicitly includes them, and all boundary problems can be formulated with arbitrary global parts in their boundary conditions. Those so-called Stieltjes boundary conditions turn out to be the right ideal $|\Phi)$ generated by the characters Φ on $\mathcal{F}$. For the case of ordinary differential equations (*assuming* that we can solve the underlying homogeneous differential equation), we arrive then at the following constructive version of the results in Section 5.

Definition 16. An *boundary problem* over an integro-differential algebra $\mathcal{F}$ is given by a pair $(T, \mathcal{B})$ with differential operator $T \in \mathcal{F}[\partial]$ and boundary space $\mathcal{B} = [\beta_1, \ldots, \beta_n]$ generated by Stieltjes conditions $\beta_i \in |\Phi)$.

Theorem 17. *Every regular boundary problem over an integro-differential algebra $\mathcal{F}$ has a Green's operator in $\mathcal{F}[\partial, \int]$, which can be determined algorithmically.*

Here $\mathcal{F}[\partial]$ denotes the usual subalgebra of differential operators with coefficients in $\mathcal{F}$. There is an integro analog, the algebra $\mathcal{F}[\int]$ of *integral operators* generated by $\int$ and the elements of $\mathcal{F}$; its normal forms are linear combinations of $f \int \tilde{f}$ with $f, \tilde{f} \in \mathcal{F}$. As one can show (Rosenkranz and Regensburger, 2008b, Prop 17), there is a direct decomposition $\mathcal{F}[\partial, \int] = \mathcal{F}[\partial] + \mathcal{F}[\int] + (\Phi)$ where (Φ) is the two-sided ideal generated by the characters Φ. If the boundary conditions in $(T, \mathcal{B})$ are of the usual two-point type (no global parts and derivatives of order smaller than that of T), then we can go beyond the statement of Theorem 17: In that case, $(T, \mathcal{B})^{-1} \in \mathcal{F}[\int]$, so the Green's operator is really an integral operator as one expects (and one can easily extract the corresponding Green's function).

Also the *factorization* is fully constructive in the case of ordinary differential equations (again assuming that we can solve the underlying homogeneous differential equations).

Theorem 18. *Given a regular boundary problem* $(T, \mathcal{B})$ *over an integro-differential algebra* $\mathcal{F}$*, every factorization* $T = T_1T_2$ *of the differential operator can be lifted algorithmically to a factorization* $(T, \mathcal{B}) = (T_1, \mathcal{B}_1) \cdot (T_2, \mathcal{B}_2)$ *of regular boundary problems over* $\mathcal{F}$*.*

This statement includes the assertion that there is at least one choice of $\mathcal{B}_2$ such that the *right problem is regular* (then the left problem is as well). As stated before, there is a lot of choice for $\mathcal{B}_2$, and it takes some "bad luck" to hit a $\mathcal{B}_2$ such that $(T_2, \mathcal{B}_2)$ is singular. Using the simple regularity criterion mentioned after Definition 4, this can be avoided easily.

We are striving toward analogous results for (some classes) of *partial differential equations*. But this will need considerably more effort since in this case it is not yet clear how to set up the operator ring. The construction of $\mathcal{F}[\partial_x, \partial_y, \int^x, \int^y]$ is just a first attempt. It is not yet clear how the appropriate language of boundary conditions looks like, in other words: what is the proper analog of the Stieltjes conditions? This promises to be an interesting and fertile ground of future research.

6. Conclusion

There is plenty of work left in the symbolic analysis of linear boundary problems (to say nothing of the nonlinear case). We have already mentioned some of the issues with partial differential equations. Both for $\mathcal{F}[\partial, \int]$ and $\mathcal{F}[\partial_x, \partial_y, \int^x, \int^y]$, there is one very natural question that we would like to pose as a challenge: Taking $\mathcal{F}$ to be the standard examples $\mathcal{S} = C^\infty(\mathbb{R})$ and $\mathcal{T} = C^\infty(\mathbb{R} \times \mathbb{R})$, respectively, can we prove that the collection of rewrite rules is *complete*?

Let us make this question precise. For $\mathcal{F}[\partial, \int]$, we have presented the "complete" collection of rewrite rules in Rosenkranz and Regensburger (2008b, Section 3). In the case of $\mathcal{F}[\partial_x, \partial_y, \int^x, \int^y]$, we have sketched some rewrite rules in Rosenkranz et al. (2009); we will present a "complete" collection in a forthcoming article. These purportedly complete collections, dubbed *Green's system* in Rosenkranz (2005), are indeed complete iff the quotient algebra $\mathcal{F}[\partial, \int]$ and $\mathcal{F}[\partial_x, \partial_y, \int^x, \int^y]$ is isomorphic to the *operator algebra* over $\mathcal{S}$ and $\mathcal{T}$, respectively. Here the operator algebra over an integro-differential algebra $\mathcal{F}$ is defined as the subalgebra of the algebra of K-linear operators on $\mathcal{F}$ that is generated by the corresponding basic operators: derivation(s), integral(s), multiplication operators $f \in \mathcal{F}$, the chosen characters $\varphi \in \Phi$, and in the partial case also the substitution operators.

We can also phrase the question in terms of the corresponding *relation ideals*: In the operator algebra, this is the ideal $\mathcal{R}$ of all relations between the basic operators; in $\mathcal{F}[\partial, \int]$ and $\mathcal{F}[\partial_x, \partial_y, \int^x, \int^y]$ it is the ideals $\mathcal{G}$ generated by the Green's systems. Now we want these ideals to be the "same" (up to isomorphism, that it): We need $\mathcal{G} \subseteq \mathcal{R}$ for correctness and $\mathcal{R} \subseteq \mathcal{G}$ for completeness.

Acknowledgements

The authors would like to thank the anonymous referees for various valuable suggestions. In particular, the encouragement for concrete examples has proved to be fertile ground for new insight and illumination. The first author acknowledges partial support by the Austrian Science Foundation (FWF), Doctorate College Computational Mathematics at Johannes Kepler University, Linz, Austria.

References

Albrecher, H., Constantinescu, C., Pirsic, G., Regensburger, G., Rosenkranz, M., 2010. An algebraic operator approach to the analysis of Gerber-Shiu functions. Insurance Math. Econom. 46, 42–51.

Baxter, G., 1960. An analytic problem whose solution follows from a simple algebraic identity. Pacific J. Math. 10, 731–742.

Bronstein, M., 2005. Symbolic Integration. I, 2nd edition. In: Algorithms and Computation in Mathematics, vol. 1. Springer-Verlag, Berlin, transcendental functions, With a foreword by B.F. Caviness.

Buchberger, B., 1965. An algorithm for finding the bases elements of the residue class ring modulo a zero dimensional polynomial ideal (German). Ph.D. thesis, Univ. of Innsbruck. English translation published in J. Symbolic Comput. 41(3–4) (2006) 475–511.

Buchberger, B., 1970. Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems. Aequationes Math. 4, 374–383. English translation: An algorithmical criterion for the solvability of a system of algebraic equations. In: B. Buchberger, F. Winkler (Eds.), Gröbner Bases and Applications, Cambridge University Press, 1998.

Buchberger, B., 1998. Introduction to Gröbner bases. In: Buchberger, B., Winkler, F. (Eds.), Gröbner Bases and Applications. Linz, 1998. Cambridge Univ. Press.

Buchberger, B., 2001. Gröbner rings and modules. In: Maruster, S., Buchberger, B., Negru, V., Jebelean, T. (Eds.), Proceedings of SYNASC 2001. The 3rd International Workshop on Symbolic and Numeric Algorithms for Scientific Computing, University of the West, Timisoara, Romania, 2–5 October. pp. 22–25. rISC-Linz Report Series No. 01-20.

Buchberger, B., 2008. Gröbner Bases in theorema using functors. In: Faugere, J., Wang, D. (Eds.), SCC'08. Proceedings of the First International Conference on Symbolic Computation in Cryptography, Beijing, China, April 28–30, 2008. LMIB Beihang University Press, pp. 1–15.

Buchberger, B., Craciun, A., Jebelean, T., Kovacs, L., Kutsia, T., Nakagawa, K., Piroi, F., Popov, N., Robu, J., Rosenkranz, M., Windsteiger, W., 2006. Theorema: towards computer-aided mathematical theory exploration. J. Appl. Log. 4 (4), 359–652. iSSN 1570-8683.

Chyzak, F., 1994. Holonomic systems and automatic proofs of identities. Tech. Rep. 2371, Institut National de la Récherche en Informatique et Automatique.

Cucker, F., Shub, M. (Eds.), 1997. Foundations of Computational Mathematics. Springer. See http://www.focm.net/ for other FoCM based publications.

Duffy, D.G., 2001. Green's Functions with Applications. In: Studies in Advanced Mathematics, Chapman & Hall/CRC, Boca Raton, FL.

Evans, L.C., 1998. Partial Differential Equations. In: Graduate Studies in Mathemtics, American Mathematical Society, Providence, Rhode Island.

Gerdt, V.P., 1999. Completion of linear differential systems to involution. In: Computer Algebra in Scientific Computing—CASC'99. Munich. Springer, Berlin, pp. 115–137.

Grabmeier, J.E., Kaltofen, E.E., Weispfenning, V.E., 2003. Computer Algebra Handbook. Foundations, Applications, Systems. With CD-ROM, demo versions. Springer, Berlin.

Guo, L., 2002. Baxter algebras and differential algebras. In: Differential Algebra and Related Topics. Newark, NJ, 2000. World Sci. Publ., River Edge, NJ, pp. 281–305.

Guo, L., Keigher, W., 2008. On differential Rota-Baxter algebras. J. Pure Appl. Algebra 212 (3), 522–540.

Helton, J.W., Wavrik, J.J., 1994. Rules for computer simplification of the formulas in operator model theory and linear systems. In: Nonselfadjoint Operators and Related Topics. Beer Sheva, 1992. In: Oper. Theory Adv. Appl., vol. 73. Birkhäuser, Basel, pp. 325–354.

Keigher, W.F., 1997. On the ring of Hurwitz series. Comm. Algebra 25 (6), 1845–1859.

Korporal, A., Regensburger, G., Rosenkranz, M., 2011. Regular and singular boundary problems in maple. In: Proceedings of the 13th International Workshop on Computer Algebra in Scientific Computing. CASC'2011, Kassel, Germany, September 5–9, 2011. In: Lecture Notes in Computer Science, vol. 6885. Springer.

Milner, R., Tofte, M., Harper, R., MacQueen, D., 1997. The Definition of Standard ML, revised edition. MIT Press.

Mora, F., 1986. Groebner bases for non-commutative polynomial rings. In: AAECC-3: Proceedings of the 3rd International Conference on Algebraic Algorithms and Error-Correcting Codes. Springer-Verlag, London, UK, pp. 353–362.

Mora, T., 1994. An introduction to commutative and noncommutative Gröbner bases. In: second International Colloquium on Words, Languages and Combinatorics, Kyoto, 1992. Theoret. Comput. Sci. 134 (1), 131–173.

Olver, P.J., 1993. Applications of Lie Groups to Differential Equations, 2nd edition. In: Graduate Texts in Mathematics, vol. 107. Springer, New York.

Plesken, W., Robertz, D., 2010. Linear differential elimination for analytic functions. Math. Comput. Sci. 4, 231–242. doi:10.1007/s11786-010-0053-2.

Regensburger, G., Rosenkranz, M., 2009a. An algebraic foundation for factoring linear boundary problems. Ann. Mat. Pura Appl. (4) 188 (1), 123–151. doi:10.1007/s10231-008-0068-3.

Regensburger, G., Rosenkranz, M., 2009b. Symbolic Integral Operators and Boundary Problems. In: Lecture Notes.

Regensburger, G., Rosenkranz, M., Middeke, J., 2009. A skew polynomial approach to integro-differential operators. In: ISSAC 2009—Proceedings of the 2009 International Symposium on Symbolic and Algebraic Computation. ACM, New York, pp. 287–294.

Renardy, M., Rogers, R.C., 2004. An Introduction to Partial Differential Equations, 2nd edition. In: Texts in Applied Mathematics, vol. 13. Springer-Verlag, New York.

Risch, R.H., 1969. The problem of integration in finite terms. Trans. Amer. Math. Soc. 139, 167–189.

Rosenkranz, M., 2003. The Green's algebra: A polynomial approach to boundary value problems. Phd thesis, Johannes Kepler University, Research Institute for Symbolic Computation, also available as RISC Technical Report 03–05, July 2003.

Rosenkranz, M., 2005. A new symbolic method for solving linear two-point boundary value problems on the level of operators. J. Symbolic Comput. 39 (2), 171–199.

Rosenkranz, M., Buchberger, B., Engl, H.W., 2003. Solving linear boundary value problems via non-commutative Gröbner bases. Appl. Anal. 82, 655–675.

Rosenkranz, M., Regensburger, G., 2008a. Integro-differential polynomials and operators. In: Jeffrey, D. (Ed.), ISSAC'08: Proceedings of the 2008 International Symposium on Symbolic and Algebraic Computation. ACM Press.

Rosenkranz, M., Regensburger, G., 2008b. Solving and factoring boundary problems for linear ordinary differential equations in differential algebras. J. Symbolic Comput. 43 (8), 515–544.

Rosenkranz, M., Regensburger, G., Tec, L., Buchberger, B., 2009. A symbolic framework for operations on linear boundary problems. In: Gerdt, V.P., Mayr, E.W., Vorozhtsov, E.H. (Eds.), Computer Algebra in Scientific Computing. Proceedings of the 11th International Workshop, CASC 2009. In: LNCS, vol. 5743. Springer, Berlin, pp. 269–283.

Rosenkranz, M., Regensburger, G., Tec, L., Buchberger, B., 2011. Symbolic analysis of boundary problems: from rewriting to parametrized gröbner bases. In: Langer, U., Paule, P. (Eds.), Numerical and Symbolic Scientific Computing. Springer, p. TBC.

Rota, G.-C., 1969. Baxter algebras and combinatorial identities (I, II). Bull. Amer. Math. Soc. 75, 325–334.

Salvy, B., Zimmerman, P., 1994. Gfun: a maple package for the manipulation of generating and holonomic functions in one variable. ACM Trans. Math. Softw. 20 (2), 163–177.

Seiler, W., 1997. Computer algebra and differential equations. An Overview mathPAD (7), 34–49.

Stakgold, I., 2000. Boundary value Problems of Mathematical Physics. Vol. I, II. In: Classics in Applied Mathematics, vol. 29. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, corrected reprint of the 1967–68 original.

Tec, L., 2011. A symbolic framework for general polynomial domains in theorema: Applications to boundary problems. Ph.D. thesis, Research Institute for Symbolic Computation, Johannes Kepler University, A-4040 Linz, Austria.
Tomuţa, E., 1998. Functor programming in theorema. Ph.D. thesis, Research Institute for Symbolic Computation, Johannes Kepler University, A-4040 Linz, Austria.
Ufnarovski, V., 1998. Introduction to noncommutative Gröbner bases theory. In: Buchberger, B., Winkler, F. (Eds.), Gröbner Bases and Applications. Linz, 1998. Cambridge Univ. Press.
van der Put, M., Singer, M.F., 2003. Galois Theory of Linear Differential Equations. In: Grundlehren der Mathematischen Wissenschaften, vol. 328. Springer, Berlin.
Wu, W.-T., 2008. Selected Works of Wen-Tsun Wu. World Scientific Publishing Company, iSBN 9812791078, 467 pages.
Yosida, K., 1995. Functional Analysis. Classics in Mathematics. Springer-Verlag, Berlin, reprint of the sixth (1980) edition.